Leveraging Terrier v4.0 for New Search Scenarios, from Smart Cities to Medical Search

Iadh Ounis, Craig Macdonald, Richard McCreadie

Search Solutions 2014 – 27th November 2014
Terrier is an open source Information Retrieval platform that facilitates...

- Efficient handling of massive-scale corpora
- Support for different types of languages
- Effective tackling of complex search tasks
- Evaluation of multiple experimental scenarios

Terrier is like Lucene, but with a research focus

- We provide richer state-of-the-art search functionalities out-of-the-box
  - We aim to be ahead of the curve in terms of features for making effective results
- Everything is configurable and can be easily changed
- .... but we provide less support for deployment (we have no or ElasticSearch equivalent)
Timeline

**2001**: Project funded to develop an IR system in Java

**2003**: Terrier 1.0: 1st open source release

**2005**: Terrier 1.1: UTF/multi-lingual support

**2007**: Terrier 2.0: Single-pass indexing support for larger corpora

**2009**: Terrier 2.2: Distributed MapReduce Support

**2011**: Terrier 3.0: Field-based and proximity models

**2013**: Terrier 3.5: DAAT, Query-biased Snippets

**2014**: Terrier 4.0: Real-time Indexing, Learning to Rank
Terrier 4.0 was released on the 18th of June 2014 and added:

- **Built-in Learning-to-Rank Support**: Learning-to-rank approaches to provide state-of-the-art effectiveness.

- **Real-time Index Structures**: Fast incremental indexing for real-time search tasks.

- **Pluggable Compression**: Configurable compression of disk-based indices (e.g. PFORDelta).

- **Website Search**: An example of online crawling, indexing and search application.

- **Index Structure Refactoring**: Extensible index structures.

- **Multi-Index**: A generic index wrapper that enables multiple indices to be searched at once.
Learning To Rank

The search models that are deployed by platforms like Lucene and Indri use one feature – the document weighting model

- (possibly mixed with a few other techniques, such as proximity search)

State-of-the-art search engines tend to use supervised models

- The main advance in IR over the last few years has been the introduction of learning to rank approaches that combine 10s to 1000s of features

<table>
<thead>
<tr>
<th>Type</th>
<th>Evidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Query</td>
<td>Query category, query difficulty</td>
</tr>
<tr>
<td>Document</td>
<td>Retrieval score, PageRank, anchor text relevance</td>
</tr>
<tr>
<td>User</td>
<td>Search personalisation</td>
</tr>
<tr>
<td>Time</td>
<td>Current stories, popular clicks</td>
</tr>
<tr>
<td>Past Queries</td>
<td>Past documents clicked for the query</td>
</tr>
</tbody>
</table>

Terrier 4.0 integrates learning to rank techniques that build more effective ranking models from training examples and features
**Learning to Rank Process**

### Training examples

- **Query Dependant**, e.g. BM25, DPH
- **Query Independent**, e.g. PageRank

<table>
<thead>
<tr>
<th>Document Sample</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1$</td>
<td>0.2</td>
<td>0.1</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td>$d_2$</td>
<td>0.5</td>
<td>0.7</td>
<td>0.0</td>
<td>0.2</td>
</tr>
<tr>
<td>$d_3$</td>
<td>0.3</td>
<td>0.2</td>
<td>0.5</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Learning to Rank Process

Training examples

document sample
Learning to Rank Process

Training examples

document sample

unseen query

Learner

Learned Model

Terrier v4.0 includes the Jforest library
...with state-of-the-art learners such as LambdaMART
Applications

Leaning to Rank approaches can be effective for different applications, e.g.

**Medical Search**

- **Doctor**
  - **query**
  - ‘temporomandibular joint dysfunction’

- **Diagnosis?**
- **Treatments?**
- **In children?**
- **Case studies?**

(Pain in the muscles that move the jaw)

Image: TRIPDatabase.com
Applications

Leaning to Rank approaches can be effective for different applications, e.g.

**Medical Search**
- Doctor
  - query: ‘temporomandibular joint dysfunction’
  - Diagnosis?
  - Treatments?
  - In children?
  - Case studies?

**Venue Suggestion**
- Entertain me!
- Location: Springfield
- How do we rank these venues?

(Pain in the muscles that move the jaw)
**Real-time Index Structures**

Classical indexing approaches are batch-orientated

- The entire collection is indexed in one go
- Documents cannot be retrieved until all documents have been indexed
Real-time Index Structures

Classical indexing approaches are batch-orientated
- The entire collection is indexed in one go
- Documents cannot be retrieved until all documents have been indexed

Terrier 4.0 introduces real-time indexing that enables parallel indexing and retrieval
- Documents are searchable as soon as they are indexed
- Enables search in new domains:
Terrier can be used for a variety of search applications

- v4.0 enables us to greatly expand this space!
“Find me more information about an event”

TOPIC TRACKING
Tracking ongoing events is a use case that search systems are increasingly being used for

- This can be seen as an incremental search task
Previously, we would have needed to build an index for each time interval before searching

- This is slow, and search effectiveness may be reduced since the collection statistics will be local to each interval
Previously, we would have needed to build an index for each time interval before searching

- This is slow, and search effectiveness may be reduced since the collection statistics will be local to each interval

Terrier 4.0’s real-time indices do this with only a single index

- If the time intervals are large, or there are many tweets, Terrier can also spread the index onto the local disk(s)
We have also distributed Terrier 4.0 search using Storm

- Index shards can be replicated to parallelise computation
- Querying happens in parallel with indexing

A related study on architecture scaling with Storm can be found in:
“Entertain Me!”

SMART CITIES AND VENUE SUGGESTION
Challenges for Information Access in Smart Cities

Searching Smart Cities
• *localised* and *timely* information needs of citizens and authorities

Events: What is happening now/today?

City Data
• *heterogeneous, various, voluminous (big data), and dynamic* sources of city data

Context & Recommendation
• *zero-query* access
• rich *contextual* information

Venues: Where to go/eat?

HCI and Applications
• interaction modalities
• visualisation of information

http://citydashboard.org/
The Venue Suggestion Task

Entertain me!

Location: Springfield

Previous venues visited:

- Elfreths Alley Museum★★★★
- Eastern State Penitentiary★★★★
- Round Guys Brewing Co★★★★
- Darlings Cafe★★★★
- Reading Terminal Market★★
- Chinatown★★
Learning to Rank Venues

We used the learning to rank support within Terrier 4.0 to make effective venue suggestions

- There are many different types of evidence we might want to use when ranking venues
- Learning to rank allows us to combine all of these features when ranking

Location: **Springfield**

Previous venues visited:

- Elfreths Alley Museum
- Eastern State Penitentiary
- Round Guys Brewing Co
- Darlings Cafe
- Reading Terminal Market
- Chinatown
Using learning to rank models can enhance the performance of venue suggestion by up to 48%

- Via a feature ablation study, we can gain new insights

<table>
<thead>
<tr>
<th></th>
<th>P@5</th>
<th>P@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial ranking [1] (LM)</td>
<td>0.2099</td>
<td>0.1910</td>
</tr>
<tr>
<td>AFS [11] (All)</td>
<td>0.3148</td>
<td>0.2874</td>
</tr>
<tr>
<td>- CITY</td>
<td>0.3058</td>
<td>0.2848</td>
</tr>
<tr>
<td>- CAT</td>
<td>0.3058</td>
<td>0.2888</td>
</tr>
<tr>
<td>- USER</td>
<td>0.3031</td>
<td>0.2794</td>
</tr>
<tr>
<td>- VENUE</td>
<td>0.3058</td>
<td>0.2744</td>
</tr>
<tr>
<td>LambdaMART [16] (All)</td>
<td>0.3713</td>
<td>0.3211</td>
</tr>
<tr>
<td>- CITY</td>
<td>0.3668</td>
<td>0.3256</td>
</tr>
<tr>
<td>- CAT</td>
<td>0.3570</td>
<td></td>
</tr>
<tr>
<td>- USER</td>
<td>0.4009</td>
<td></td>
</tr>
<tr>
<td>- VENUE</td>
<td>0.2960</td>
<td></td>
</tr>
</tbody>
</table>

A classical language modelling approach

Learning to Rank

User personalisation features are surprisingly not that useful!
Entertain Me!
Venue Suggestion on Android...

User Authentication Screen

Recommended Venues Screen

Venue details Screen

Round-up

SUMMARY
Summary

Terrier is an information retrieval platform that supports research and development of search systems across domains:

- Web Search
- News Search
- Patient Search
- Microblog Search
- Search over Sensors
- Venue Suggestion

The Terrier v4.0 release provides

- Real-time index structures
- Built-in learning to rank functionality
- Compressed index structures
Get Involved

Download for free at: Terrier.org

Engage:
- We maintain a FAQ wiki
- Ask questions on our forum
- Report issues or submit patches on our issue tracker

Collaborate:
- We welcome collaborations with industry and the public sector
  - Secondment and exchange of visits
  - Joint PhD supervision
  - KE projects

Come talk to us!
Questions?
ladh.OUNIS@Glasgow.ac.uk

Terrier.org
References

Learning to Rank


Compression


Storm Architectures


Venue Suggestion
